

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for moving data objects ~~(201.x)~~ in a computer system ~~(101)~~ from a first ~~(107)~~ storage location to a second ~~(108)~~ storage location, comprising:
 - a) selecting one or more data objects ~~(201.x)~~ from the first storage location ~~(107)~~_{i,1};
 - b) assigning at least one identifier (ID) of at least one type to each of the selected data objects ~~(201.x)~~_{i,1};
 - c) storing the ~~at least one~~ ID in a transactional type lock object ~~(204)~~_{i,1};
 - d) ~~in case step c) has been performed successfully:~~ determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing ~~said at least one~~ the ID in a permanent type lock object ~~(203)~~_{i,1};
 - e) ~~in case step d) has been performed successfully:~~ determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ~~at least one~~ ID from the transactional type lock object ~~(204)~~_{i,1};
 - f) storing ~~[[a]]~~ the data object ~~(201.x)~~, the ~~at least one~~ ID of which is contained in the permanent type lock object ~~(203)~~, at the second storage location ~~(108)~~ and assigning the second storage location ~~(108)~~ to the ~~at least one~~ ID in the permanent type lock object ~~(203)~~_{i,1};

g) deleting ~~[[a]]~~ the data object (201.x), the ~~at least one~~ ID of which is contained in the permanent type lock object (203), from ~~said~~ the first storage location (107); and

h) deleting ~~an at least one~~ the ID from the permanent type lock object (203) ~~earliest at a time at which step g) for~~ after the respective data object (201.x) assigned to that ~~at least one~~ ID has been ~~completed~~ deleted from the first storage location.

2. (Currently Amended) The method of claim 1, wherein each a data object (201.x) comprises one ~~ore~~ or more fields of one or more tables, (201, 202) and wherein the ~~at least one~~ ID comprises one or more key fields of the one or more tables (201, 202).

3. (Currently Amended) The method of claim 1 ~~or 2~~, wherein ~~in step f) the data objects (201.x) are~~ the data object is stored in one or more files and wherein an assignment of the ~~at least one~~ ID to a filename or file, in which the data object assigned to the ~~at least one~~ ID is to be stored, is stored in the permanent type lock object (203).

4. (Currently Amended) The method of claim 1, wherein ~~in step e) the IDs are~~ ID is stored in the transactional type lock object (204) ~~immediately after performing step b)~~ the step of assigning at least one identifier of at least one type to each of the selected data objects for the respective data object (201.x).

5. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 4~~, wherein in ~~step d)~~ storing said ID in the permanent type lock object further comprises storing the ~~IDs~~ ID of all selected data objects (201.x) ~~are stored in the permanent type lock object (203) before the~~ a first storing process according to step f) is started.

6. (Currently Amended) The method ~~one of claims~~ claim 1 ~~to 5~~, further comprising:

i) ~~checking before or while performing any of steps a) to d) for a data object (201.x), whether an ID for that one of the selected data object (201.x) objects has been stored in a lock object (203, 204), and if yes, the ID has been stored, skipping at least step f) for that the one of the selected data object (201.x) objects.~~

7. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 6~~, further comprising:

j) ~~checking before or while performing any of steps a) to f) for a data object (201.x), whether the data object (201.x) is contained in the second storage location (108), and if yes, the data object is contained, skipping at least step f) for that data object (201.x).~~

8. (Currently Amended) The method of claim 7, wherein said checking ~~according to step j)~~ is performed by querying a the lock object (203, 204).

9. (Currently Amended) The method of ~~one of claims~~ claim 1 ~~to 8~~, further comprising:

k) ~~in case of a failure in step f)~~ determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking[[[,]] whether the data object (201.x) assigned to the respective ID has been completely stored in the second storage location (108), and in case of no, if the

respective ID has not been completely stored, skipping at least steps g) and h) for that data object (201.x) and deleting the ID from the permanent type lock object (203).

10. (Currently Amended) The method of ~~one of claims~~ claim 1 to 9 for use in an enterprise resource planning software.

11. (Currently Amended) A computer system (101) for processing data ~~by means of or in a software application~~, comprising:

[[-]] memory (112) means for storing program instructions;

[[-]] input means (102, 104) for entering data;

[[-]] storage means (107, 108) for storing data;

[[-]] a processor (105) responsive to the program instructions, wherein the

[[-]] program instructions (111) ~~to carry out a method as of any of claims 1 to 10 if executed in the computer system~~ comprise program code means for performing a method for moving data objects in the computer system from a first storage location to a second storage location, the method comprising:

selecting one or more data objects from the first storage location;

assigning at least one identifier (ID) of at least one type to each of the selected data objects;

storing the ID in a transactional type lock object;

determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object;

determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

storing the data object, the ID of which is contained in the permanent type lock object, at the second storage location and assigning the second storage location to the ID in the permanent type lock object;

deleting the data object, the ID of which is contained in the permanent type lock object, from the first storage location; and

deleting the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location.

12. (Currently Amended) A computer readable medium comprising instructions ~~program code~~ means for performing a method ~~as of any of claims 1 to 10 if said program is executed on a computer system~~ for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting one or more data objects from the first storage location;

assigning at least one identifier (ID) of at least one type to each of the selected data objects;

storing the ID in a transactional type lock object;

determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object;

determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

storing the data object, the ID of which is contained in the permanent type lock object, at the second storage location and assigning the second storage location to the ID in the permanent type lock object;

deleting the data object, the ID of which is contained in the permanent type lock object, from the first storage location; and

deleting the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location.

13. (Cancelled)

14. (Cancelled)

15. (New) The computer readable medium of claim 12, wherein each data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

16. (New) The computer readable medium of claim 12, wherein the data object is stored in one or more files and wherein an assignment of the ID to a filename or file, in which the data object assigned to the ID is to be stored, is stored in the permanent type lock object.

17. (New) The computer readable medium of claim 12, wherein the ID is stored in the transactional type lock object after the step of assigning at least one

identifier of at least one type to each of the selected data objects for the respective data object.

18. (New) The computer readable medium of claim 12, wherein storing said ID in the permanent type lock object further comprises:

storing the ID of all selected data objects in the permanent type lock object before storing the data object at the second location.

19. (New) The computer readable medium of claim 12, further comprising:
checking whether an ID for ~~that~~ one of the selected data object objects has been stored in a lock object, and if the ID has been stored, skipping the storing of the data object, the ID of which is contained in the permanent type lock object, at the second storage location for ~~that~~ the one of the selected data object objects.

20. (New) The computer readable medium of claim 12, further comprising:
checking whether the data object is contained in the second storage location and if the data object is contained, skipping the storing of the data object, the ID of which is contained in the permanent type lock object, at the second storage location for that data object.

21. (New) The computer readable medium of claim 12, wherein the checking is performed by querying the lock object.

22. (New) The computer readable medium of claim 12, further comprising:
determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking whether the data object assigned

to the respective ID has been completely stored in the second storage location, and if the respective ID has not been completely stored, skipping the deleting of the data object, the ID of which is contained in the permanent type lock object, from the first storage location, and the deleting of the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location for that data object and deleting the ID from the permanent type lock object.

23. (New) A computer system for processing data, comprising:
- means for selecting one or more data objects from the first storage location;
 - means for assigning at least one identifier (ID) of at least one type to each of the selected data objects;
 - means for storing the ID in a transactional type lock object;
 - means for determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object;
 - means for determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;
 - means for storing the data object, the ID of which is contained in the permanent type lock object, at the second storage location and assigning the second storage location to the ID in the permanent type lock object;
 - means for deleting the data object, the ID of which is contained in the permanent type lock object, from the first storage location; and

means for deleting the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location.

24. (New) The computer system of claim 23, wherein each data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

25. (New) The computer system of claim 23, wherein the data object is stored in one or more files and wherein an assignment of the ID to a filename or file, in which the data object assigned to the ID is to be stored, is stored in the permanent type lock object.

26. (New) The computer system of claim 23, wherein the ID is stored in the transactional type lock object after the step of assigning at least one identifier of at least one type to each of the selected data objects for the respective data object.

27. (New) The computer system of claim 23, wherein the means for storing said ID in the permanent type lock object further comprises:

means for storing the ID of all selected data objects in the permanent type lock object before storing the data object at the second location.

28. (New) The computer system of claim 23, further comprising:
means for checking whether an ID for that data object has been stored in a lock object, and if the ID has been stored, skipping the storing of the data object, the ID of

which is contained in the permanent type lock object, at the second storage location for that data object.

29. (New) The computer system of claim 23, further comprising:

means for checking whether the data object is contained in the second storage location and if the data object is contained, skipping the storing of the data object, the ID of which is contained in the permanent type lock object, at the second storage location for that data object.

30. (New) The computer system of claim 23, wherein the checking is performed by querying the lock object.

31. (New) The computer system of claim 23, further comprising:

means for determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking whether the data object assigned to the respective ID has been completely stored in the second storage location, and if the respective ID has not been completely stored, skipping the deleting of the data object, the ID of which is contained in the permanent type lock object, from the first storage location, and the deleting of the ID from the permanent type lock object after the respective data object assigned to that ID has been deleted from the first storage location for that data object and deleting the ID from the permanent type lock object.